Heart Disease Prediction Project

Edward Amankwah

2019-11-28

Contents

1 Introduction

One of the major courses of death can be attributed to heart disease. Many researches are on-going globally in search of methods that will aid early detection and thereby leading to reduce heart disease related death and health cost. The cause of heart disease can be complex involving many biomedical and clinical factors. Researches usually draw inferences from these records stored in databases using statistical methods to predict the presence or absence of heart disease. However, machine learning algorithms are being used to augment patient hearth delivery, to develop customized computational models capable of analyzing and predicting the presence of heart disease [1].

This report is subdivided into introduction which includes motivation of the project and data preprocessing, methods and analysis, modelling approach, results, discussion, conclusions and references.

1.1 Motivation of Project

This project is the second part of the final capstone project to complete the HarvardX professional data science certification program. The project aims at developing machine learning algorithms capable of predicting the presence or absence of heart disease from data set obtained from the Cleveland Clinical Foundation, the Hungarian Institute of Cardiology (Budapest), the V.A Medical Center (Long Beach CA) and University Hospital Zurich (Switzerland).

1.1.1 Evaluation Metric

The evaluation metric for the algorithm performance is the Overall Accuracy. Overall accuracy is one of the simplest measures used to report the proportion of cases that are correctly predicted on a test set when the outcomes are categorical. The models will be built using training data set containing numerical heart disease features and the overall accuracy would be used to determine the quality of prediction on the testing dataset, which contains categorical target data. The best model would be determined based on the model prediction which produces the highest overall accuracy.

2 Dataset

The heart disease dataset (cleveland.data [2], hungarian.data [3], switzerland.data [4] and long-beach-va.data [5], heart-disease.names) can be downloaded from the UCI Machine Learning Repository. The database contains 76 attributes but this analysis refers to using a subset of 14 of them, including the target variable. The heart-disease.names file contains the details of attributes and variables.

2.1 Target Features

The response variable (goal) is given as a diagnose of heart disease (angiographic disease status) where any major vessel is designated as having "diameter narrowing" $< 50\%$ (value 0) indicate the absence of heart disease and "diameter narrowing " $>50\%$(value 1) indicating the presence of heart disease.

2.2 Predictor Features

The following description was obtained from the heart-disease.name file:

- Age: age in years
- Sex: gender (1 = male; 0 = female)
- CP: chest pain type – Value 1: typical angina – Value 2: atypical angina – Value 3: non-anginal pain – Value 4: asymptomatic
- Trestbps: resting blood pressure (in mm Hg on admission to the hospital)
- Chol: serum cholesterol in mg/dl
- FBS fasting blood sugar > 120 mg/dl (1 = true; 0 = false)

- RestECG: resting electrocardiographic results – Value 0: normal – Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 005 mV) – Value 2: showing probable or definite left ventricular hypertropy by Estes criteria
- Thalach: maximum heart rate achieved in beats per minute (bpm)
- Exang: exercise induced angina (1 = yes; 0 = no)
- Oldpeak: ST depression induced by exercise relative to rest
- Slope: the slope of the peak exercise ST segment – Value 1: upsloping – Value 2: flat – Value 3: down-sloping
- CA: number of major vessels (0-3) colored by fluoroscopy
- Thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

2.3 Preprocessing for Data Exploration

The heart disease data sets were read in as comma separated files setting the string values as characters. The column headers were separately read in and then combined with the datasets by rows. The string characters were eventually converted to factors.

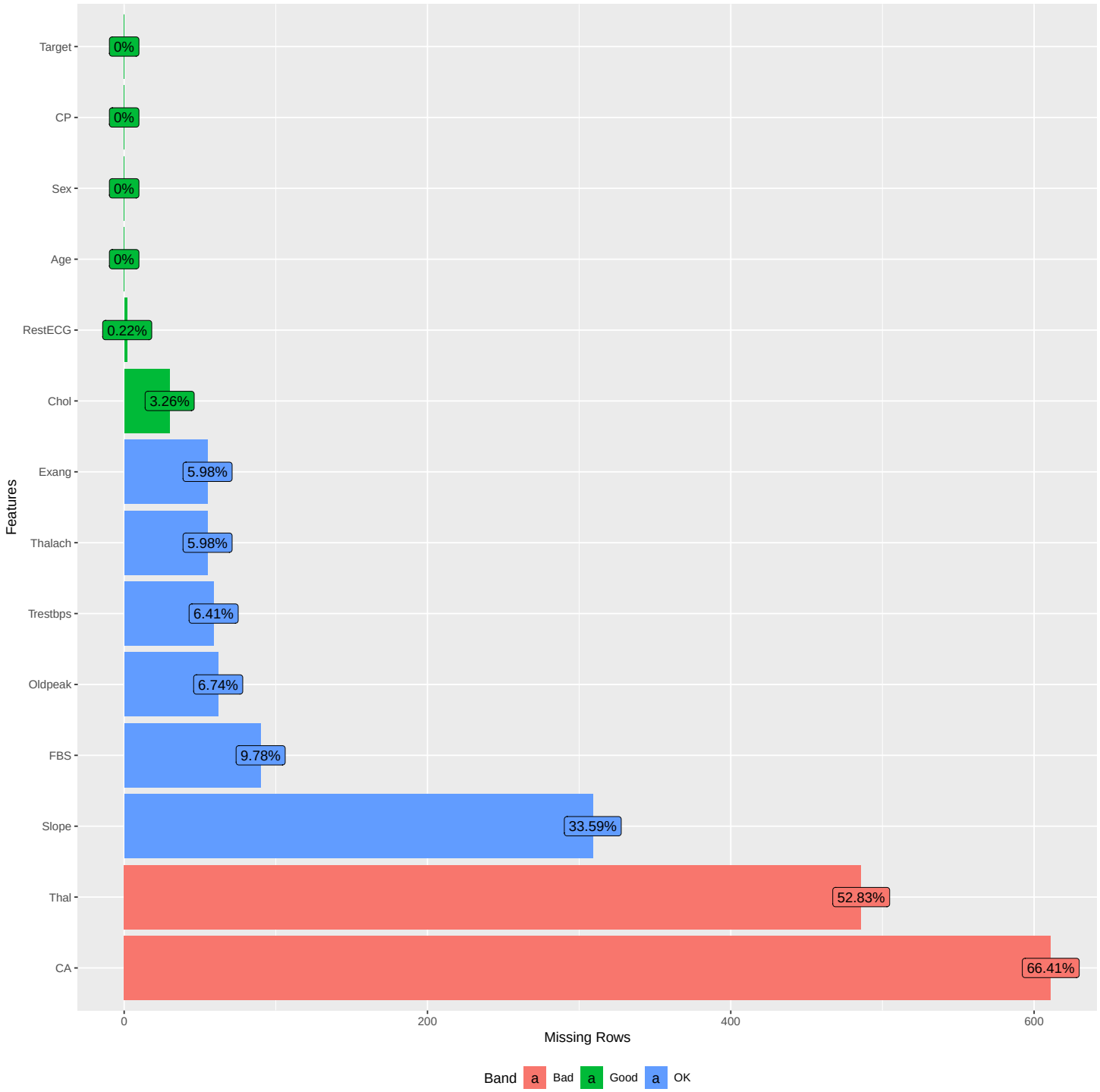
```
# Note: this process could take a couple of minutes

################################################################
# Create heart disease (hdease), training sets, and testing sets
################################################################
# Note: this process could take a couple of minutes for loading required package: tidyverse and packa
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
if(!require(mlr)) install.packages("mrl", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(caretEnsemble)) install.packages("caretEnsemble", repos = "http://cran.us.r-project.org")
if(!require(DataExplorer)) install.packages("DataExplorer", repos = "http://cran.us.r-project.org")
if(!require(MASS)) install.packages("MASS", repos = "http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.us.r-project.org")
if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(klar)) install.packages("klar", repos = "http://cran.us.r-project.org")
if(!require(xgboost)) install.packages("xgboost", repos = "http://cran.us.r-project.org")
if(!require(matrixStats)) install.packages("matrixStats", repos = "http://cran.us.r-project.org")
if(!require(fastAdaboost)) install.packages("fastAdaboost", repos = "http://cran.us.r-project.org")
if(!require(earth)) install.packages("earth", repos = "http://cran.us.r-project.org")
if(!require(ggthemes)) install.packages("ggthemes", repos = "http://cran.us.r-project.org")
if(!require(cowplot)) install.packages("cowplot", repos = "http://cran.us.r-project.org")
if(!require(Rmisc)) install.packages("Rmisc", repos = "http://cran.us.r-project.org")
options(digits = 3)

cleveland <- read.csv('processed.cleveland.csv', na = "?", stringsAsFactors = FALSE, header = FALSE)
hungarian <- read.csv('processed.hungarian.csv', na = "?", stringsAsFactors = FALSE, header = FALSE)
switzerland <- read.csv('processed.switzerland.csv', na = "?", stringsAsFactors = FALSE, header = FAI
va <- read.csv('processed.va.csv', na = "?", stringsAsFactors = FALSE, header = FALSE)
hdisease <- rbind(cleveland, hungarian, switzerland, va)
names(hdisease) <- c('Age', 'Sex', 'CP', 'Trestbps', 'Chol', 'FBS', 'RestECG',
                     'Thalach', 'Exang', 'Oldpeak', 'Slope', 'CA', 'Thal', 'Target')
```

2.4 Missing Values

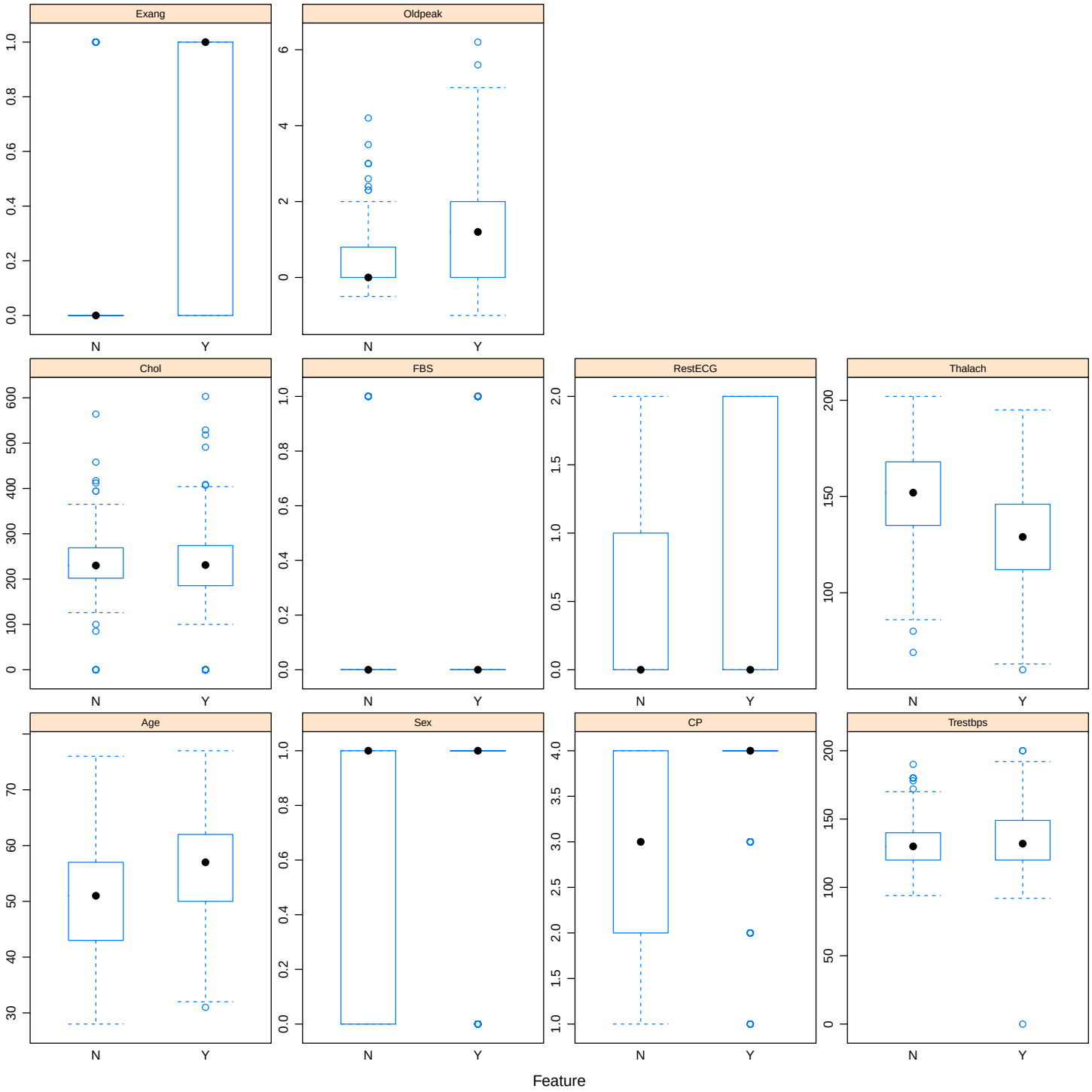
The plot below shows the distribution of missing values in the dataset.



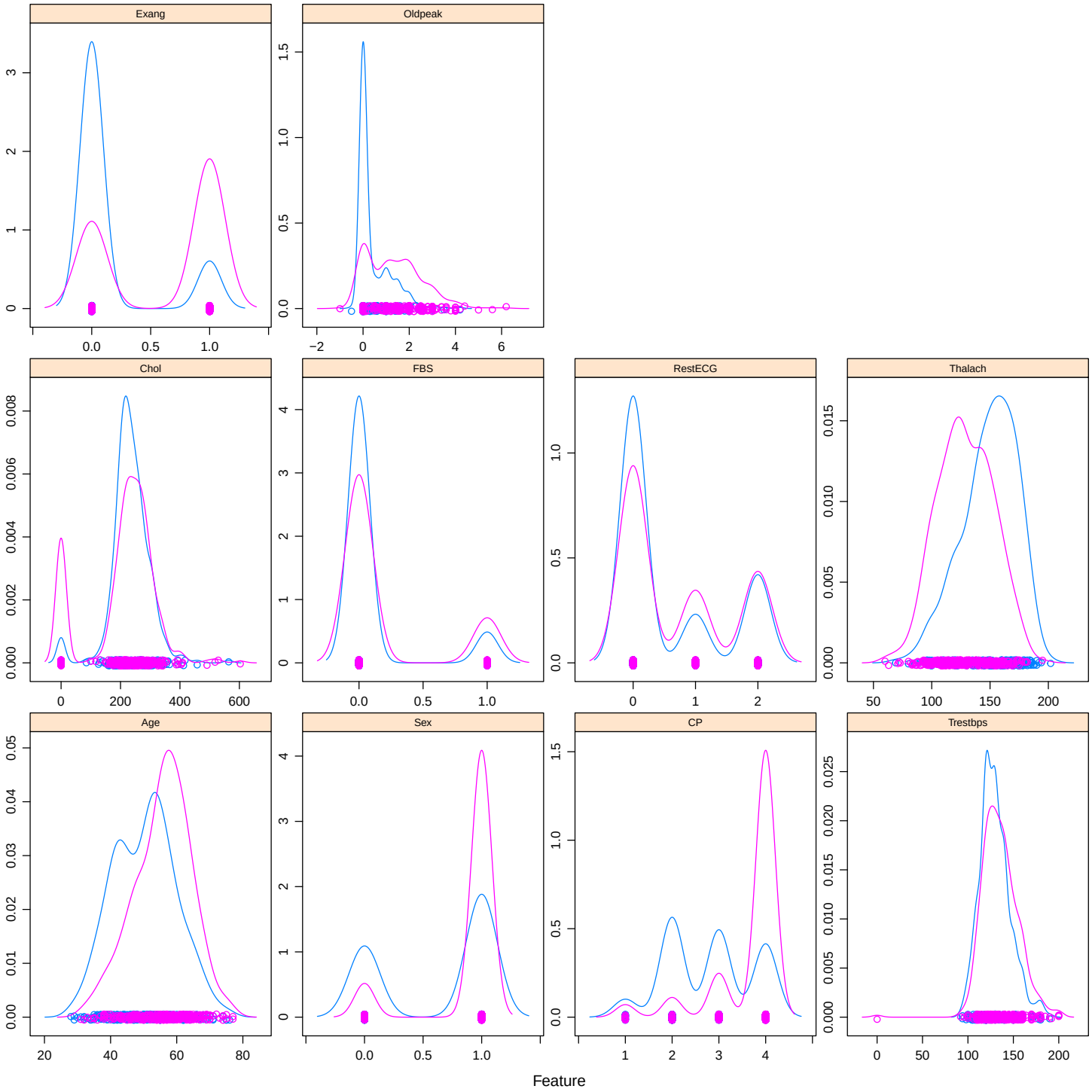
Predictive features such as slope, CA and Thal are missing more than 30% of their entire dataset.

2.5 Visualization of Important Features

After data preprocessing, it is important to visualize how the predictors will influence the response variable (Target).



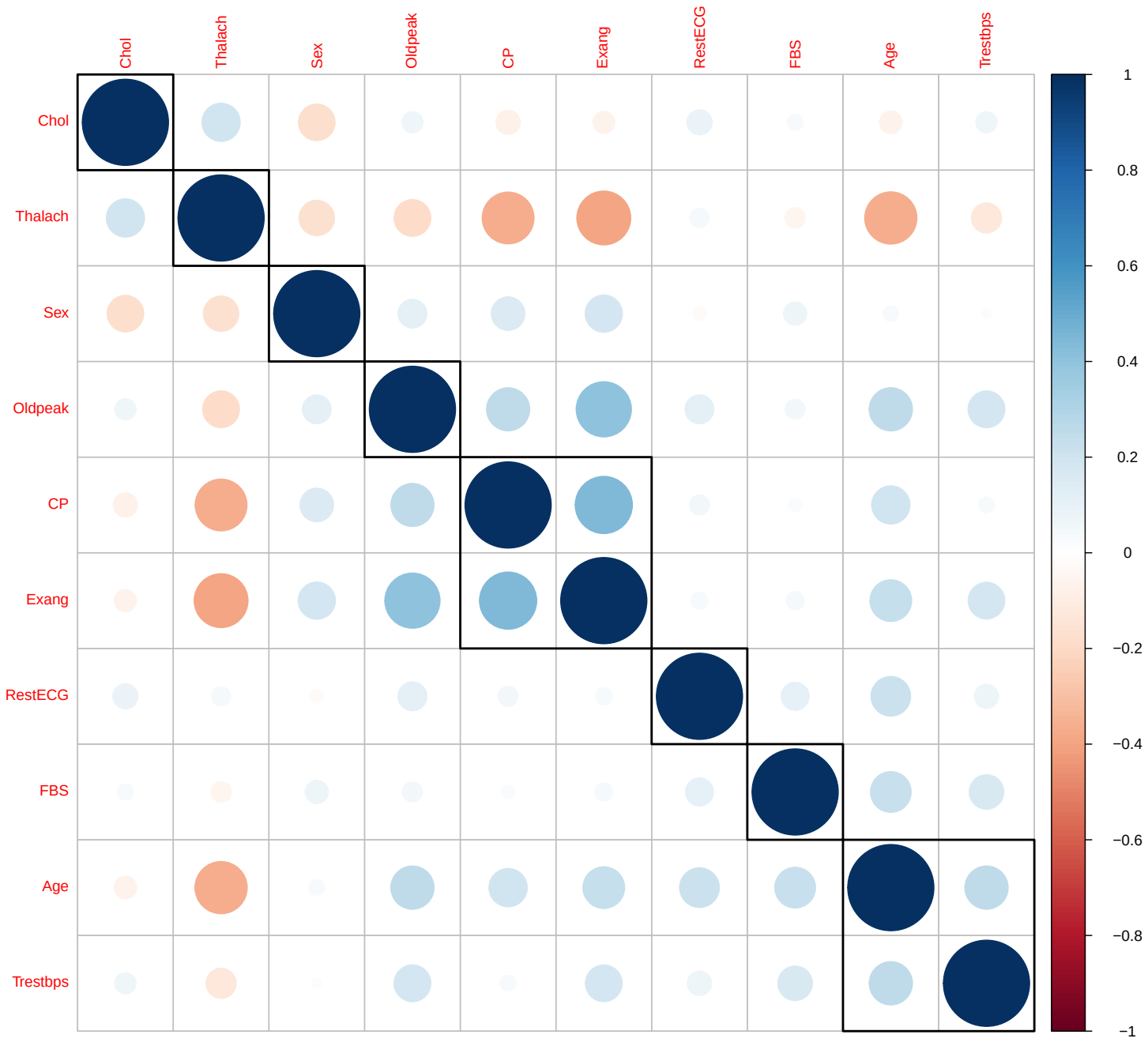
The boxplots above show features such as Oldpeak, Chol and Trestbps have a lot of extreme values outside the 25th – 75th percentiles. The mean and the placement of the two box plots in 'Oldpeak', 'CP', 'Age', and 'Thalach' are visibly different indicating that they are potential predictors of the Target feature.



Similarly, the density curves for ‘Oldpeak’, ‘CP’, ‘Age’, and ‘Thalach’ are significantly different indicating they are potential predictors of the Target feature.

3 Data Exploration

3.1 Variable Correlation Plot



Machine learning models assume the predictor variables are independent of each other. However, the plot above shows some predictors are highly correlated with others. In this analysis predictors having a degree of correlation between variables at a value greater than 0.75 (i.e. cutoff equal to 0.25) will be dropped to make the analysis more robust.

The table below shows predictors that are considered to be highly correlated and were therefore, dropped from the dataset.

Table 1: Top Correlated Features

	Age	Thalach	Exang	Oldpeak
Age	1.000	-0.367	0.234	0.252
Thalach	-0.367	1.000	-0.390	-0.182

	Age	Thalach	Exang	Oldpeak
Exang	0.234	-0.390	1.000	0.410
Oldpeak	0.252	-0.182	0.410	1.000

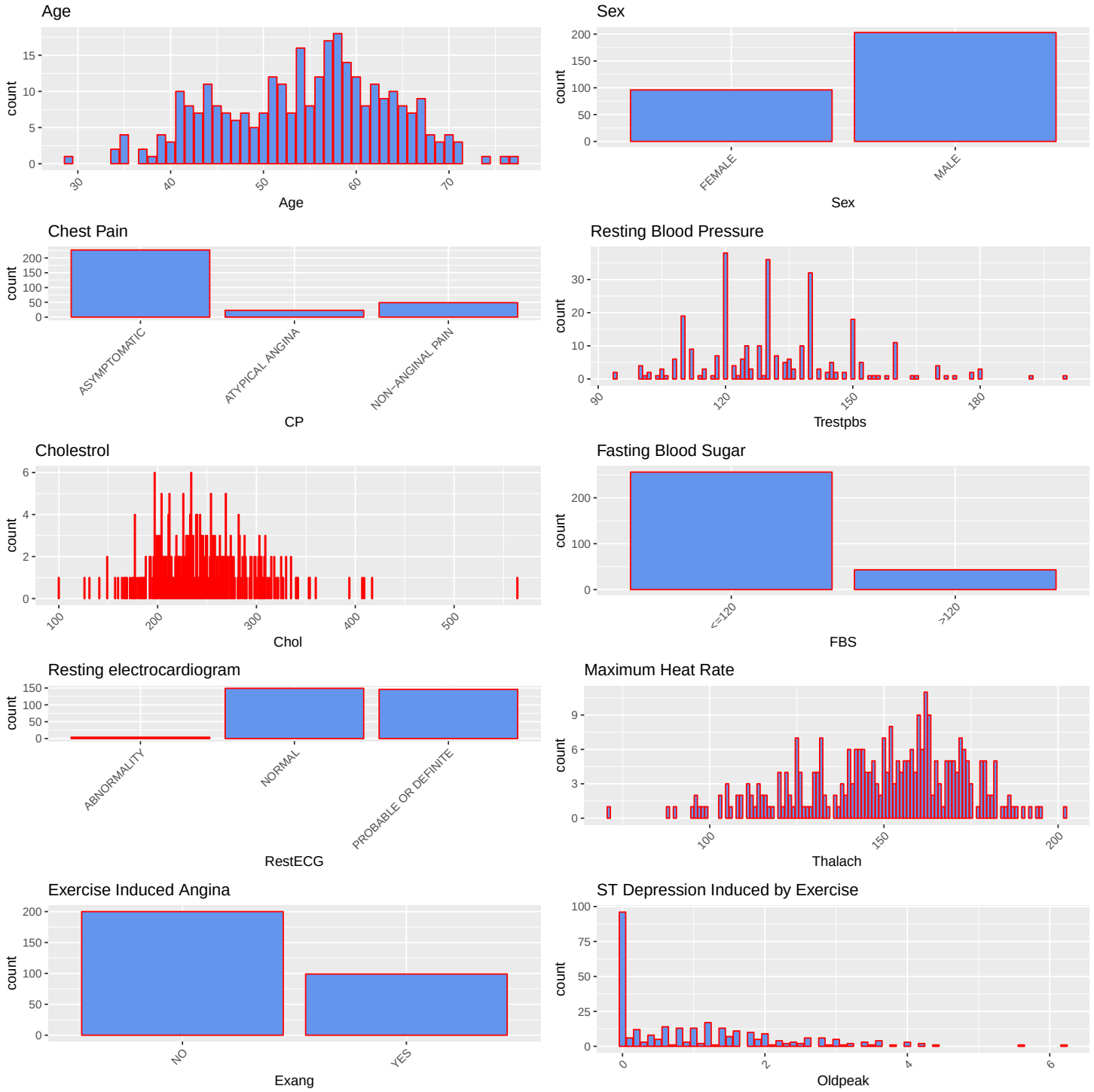
3.2 Data transformation

The following variables with more than 30% of missing values were initially dropped from the data asset because either they will be less represented or may not add significant information:

- Slope (slope of the peak exercise ST segment),
- CA (number of major vessels) and
- Thal (Thalassemia cardiomyopathy) features.

The target variable was transformed into a numerical binary outcome of whether patient has a heart disease or not (Y-1, N-0). The Sex variable was transformed into factors of Male(1) and Female(2). The fasting blood sugar (FBS) was transformed into factors of concentration > 120 mg/dl (1) and concentration <= 120 mg/dl (0). The chest pain type (CP) was transformed into factors of ATYPICAL ANGINA (1), NON-ANGINAL PAIN (2) and ASYMPTOMATIC (3). The resting electrocardiographic (RestECG) was transformed into factors of Normal (0), Abnormality(1) and Probable or Definite(2). The exercise induced angina (Exang) was transformed into factors of exercise induced angina (Y-1, N-0).

3.3 Univariate Distributions

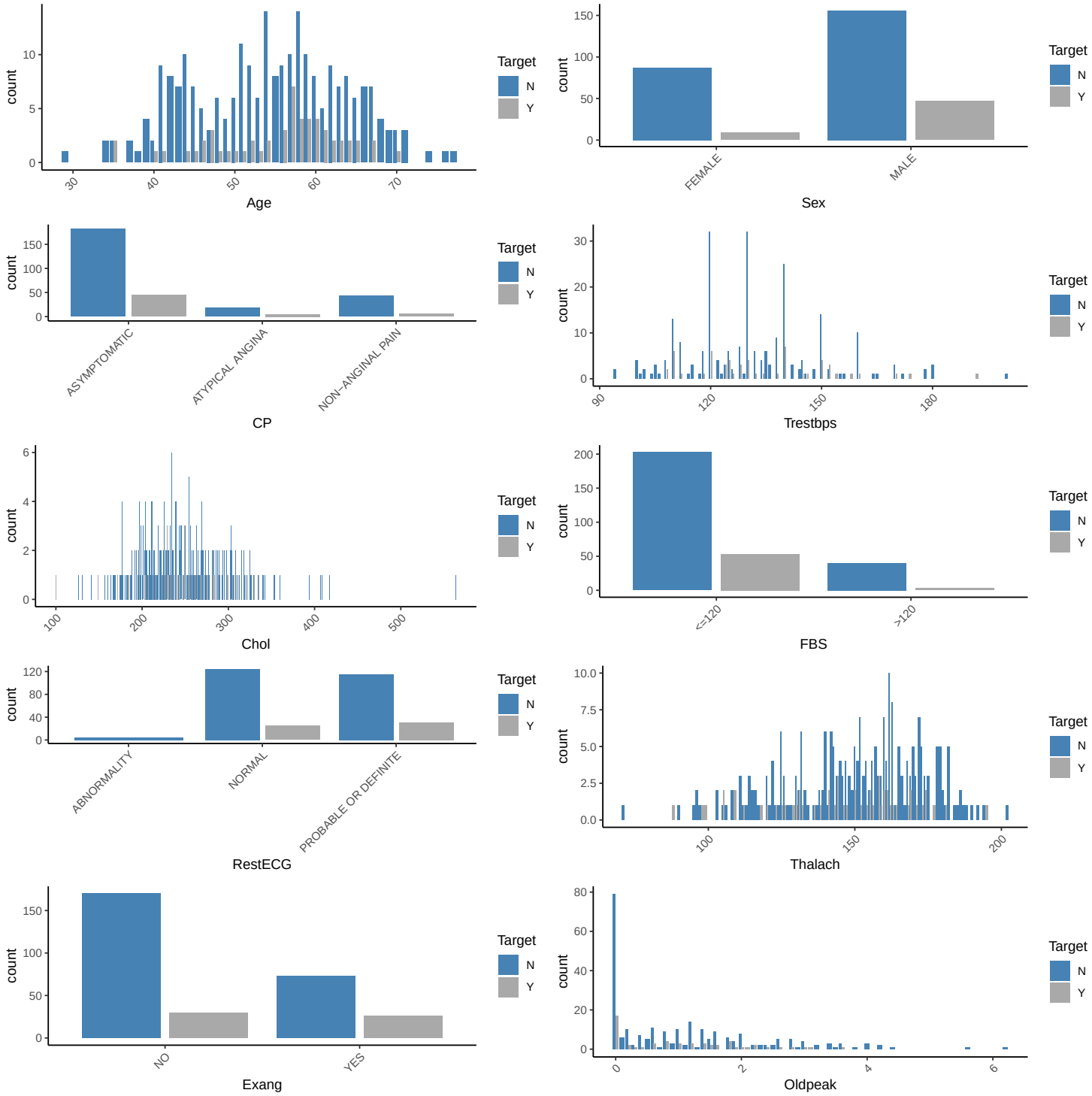


The bar plots above show that patient's ge, resting blood pressure, cholesterol, ST depression induced by excercise and maximum heart rate are somewhat normally distributed. The average age ranges between 50 and 60 years with the youngest and oldest being 28 and 77, respectively. The number of female patients are less than half of the male patients. The mean cholesterol level ranges between 200 and 300 mg/dL. The distribution of maximum heart rate is skewed towards higher beats per minute measurements. The bar plot for the ST depression induced by exercise is skewed due to the large number of low values. Fast blood sugar content greater than 120 mg/dl is significantly lower than concentrations less than 120 mg/dl. The number of patients with exercise induced angina is less than half the number of patients with no exercise induced angina.

3.4 Bivariate distribution

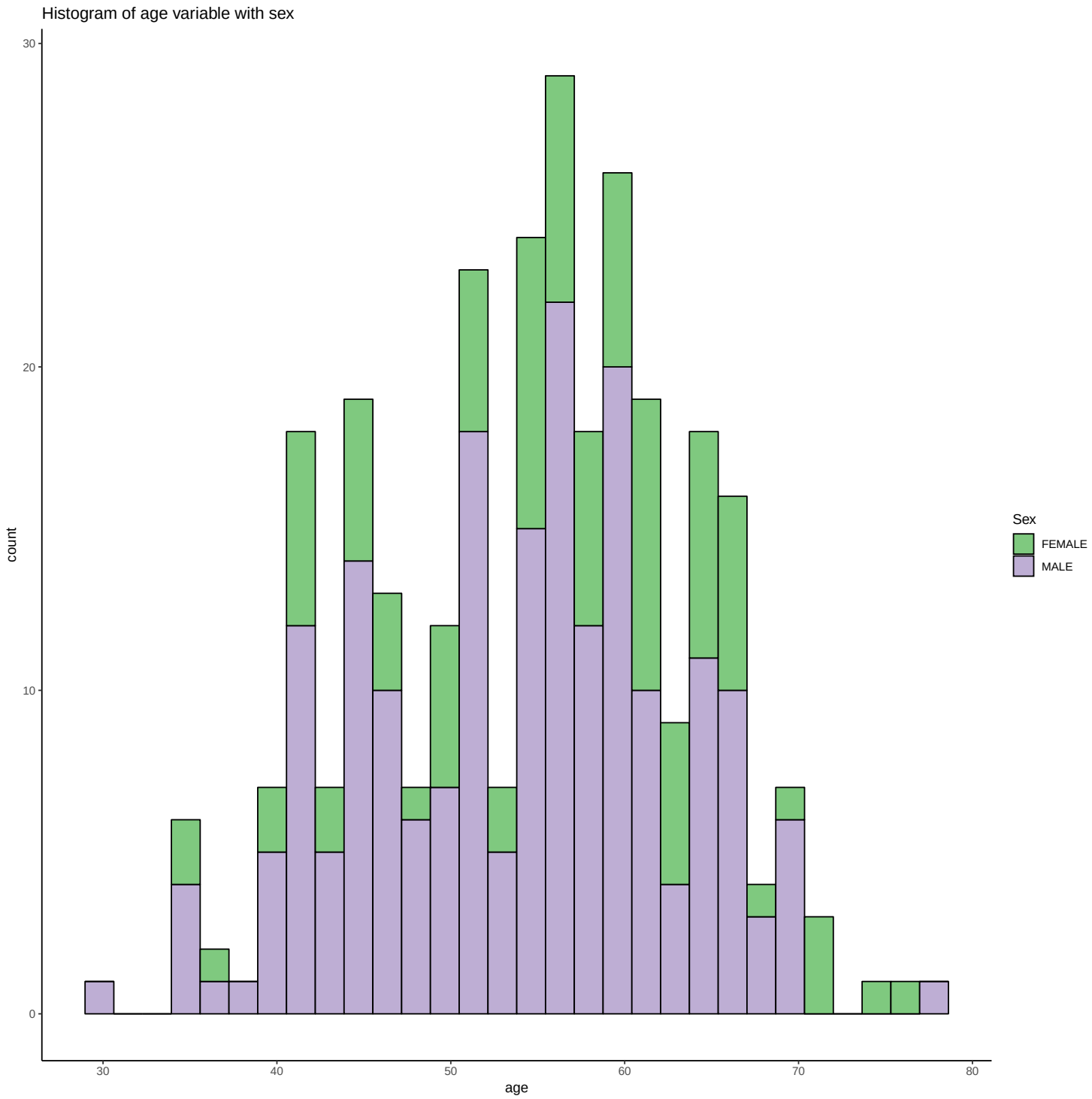
3.4.1 Predictor Variable Counts Grouped by Response Variable

The bar plots below show the count of predictor values for patients with and without heat disease.



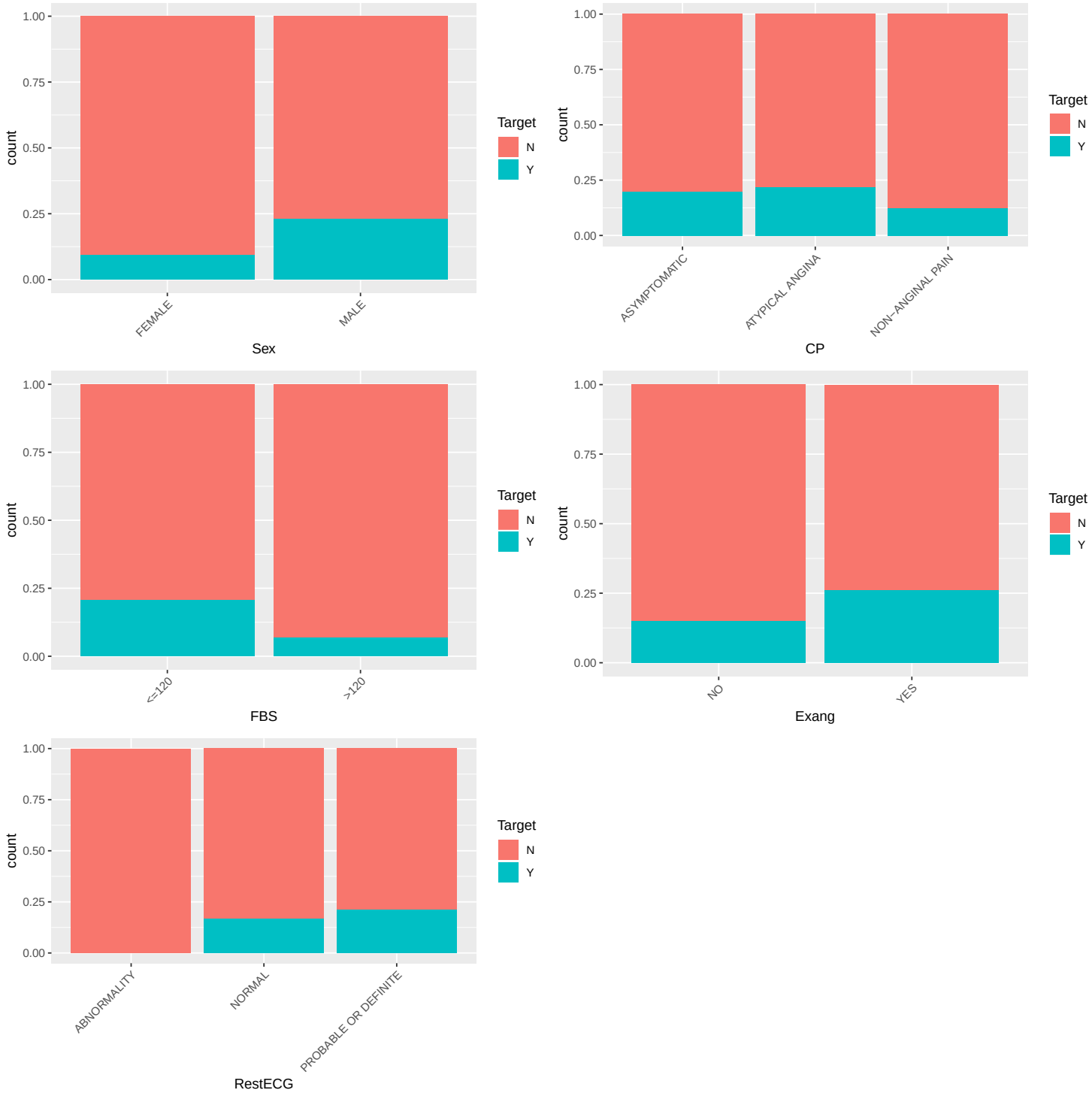
3.4.2 Variation of Sex with Age

The stack histogram below shows that more males were involved in the study than females with the oldest and youngest patients being males.


Histogram of age variable with sex
count
30
20
10
0
30
40
50
60
70
80
age
Sex
FEMALE
MALE

3.5 Categorial Features

3.5.1 Variation of Sex, Chest pain, FBS and RestECG with Target

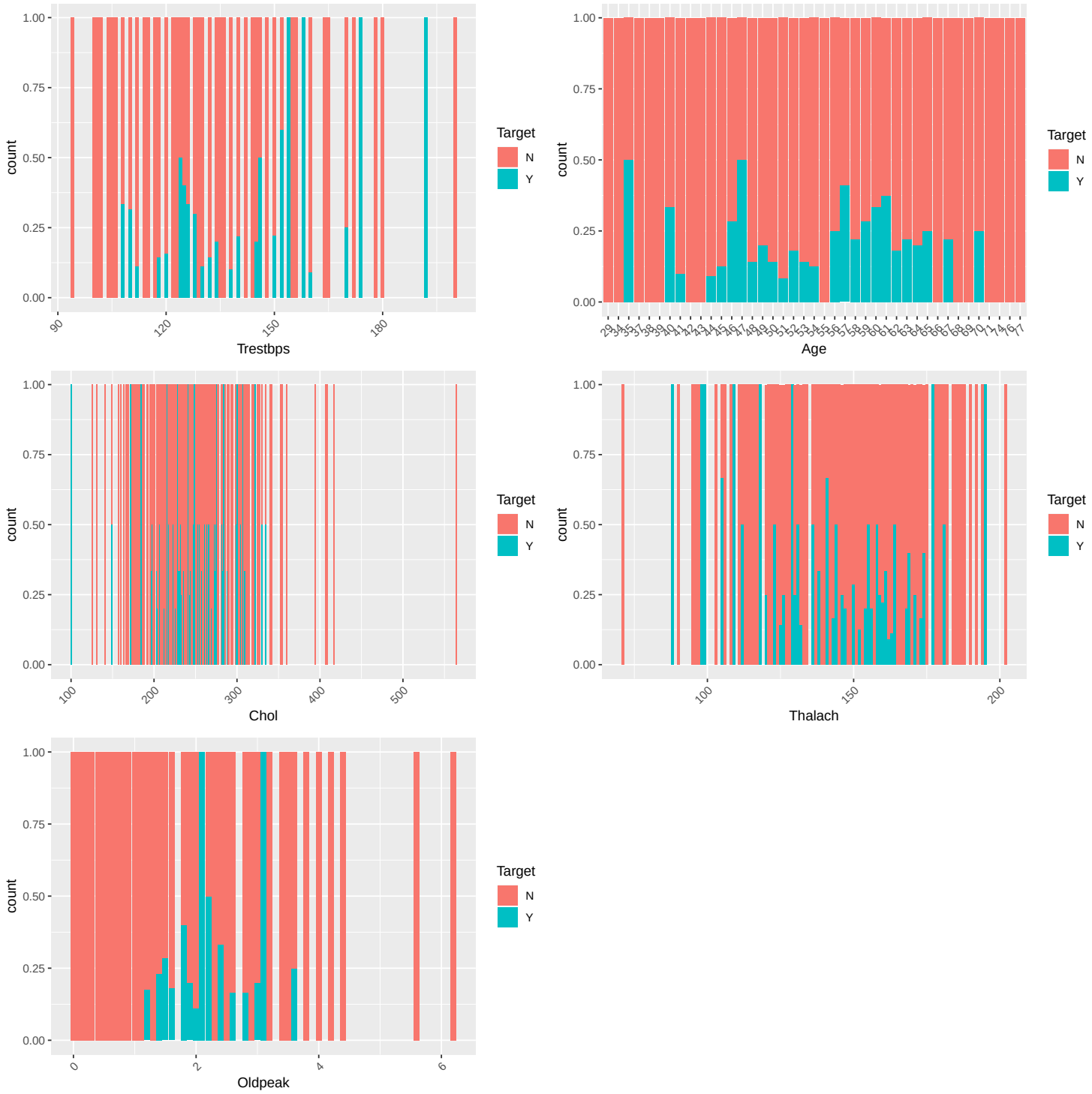


The bar plots above further buttress the fact that less than 25 % of the predictor features; sex, chest pain type, fasting blood sugar , exercise induced angina, and resting electrocardiographic indicate heart disease. Most patients didn't have fasting blood sugar levels above 120 mg/dl. Resting electrocardiographic predictor had no indication of abnormality.

3.6 Numerical Features

3.6.1 Variation of Trestbps, Age, Chol, Thalach, Exang and Oldpeak with Target

```
## Warning: position_stack requires non-overlapping x intervals
```

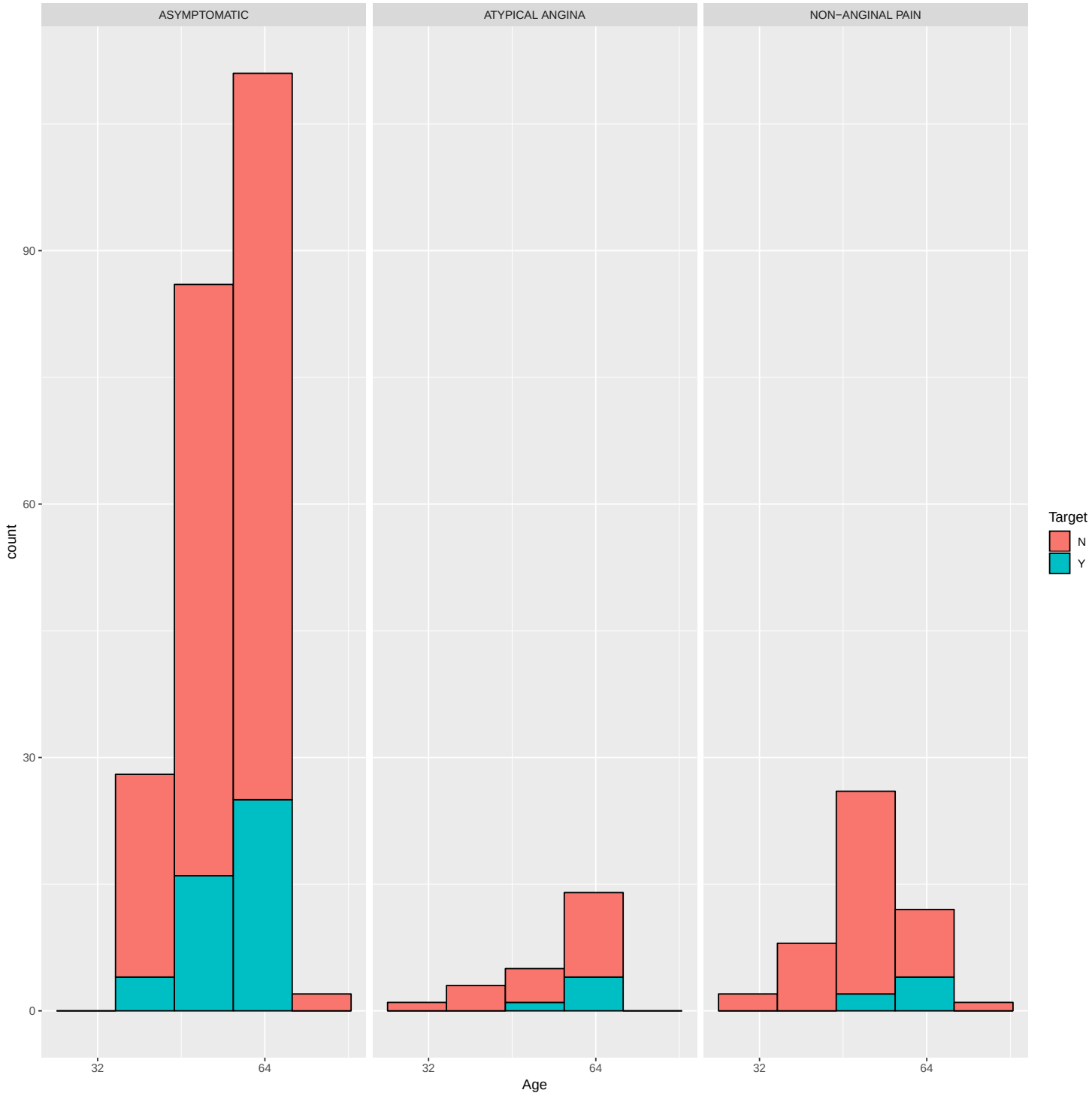


The bar plots above indicate the majority of the numerical features; trestpbs, age, cholesterol, Thalach and Oldpeak had no effect on heart disease. Most patients didn't have cholesterol levels above 350 mg/dl and ST depression induced by exercise (Oldpeak) relative to rest above 3.8.

4 Multivariate distribution

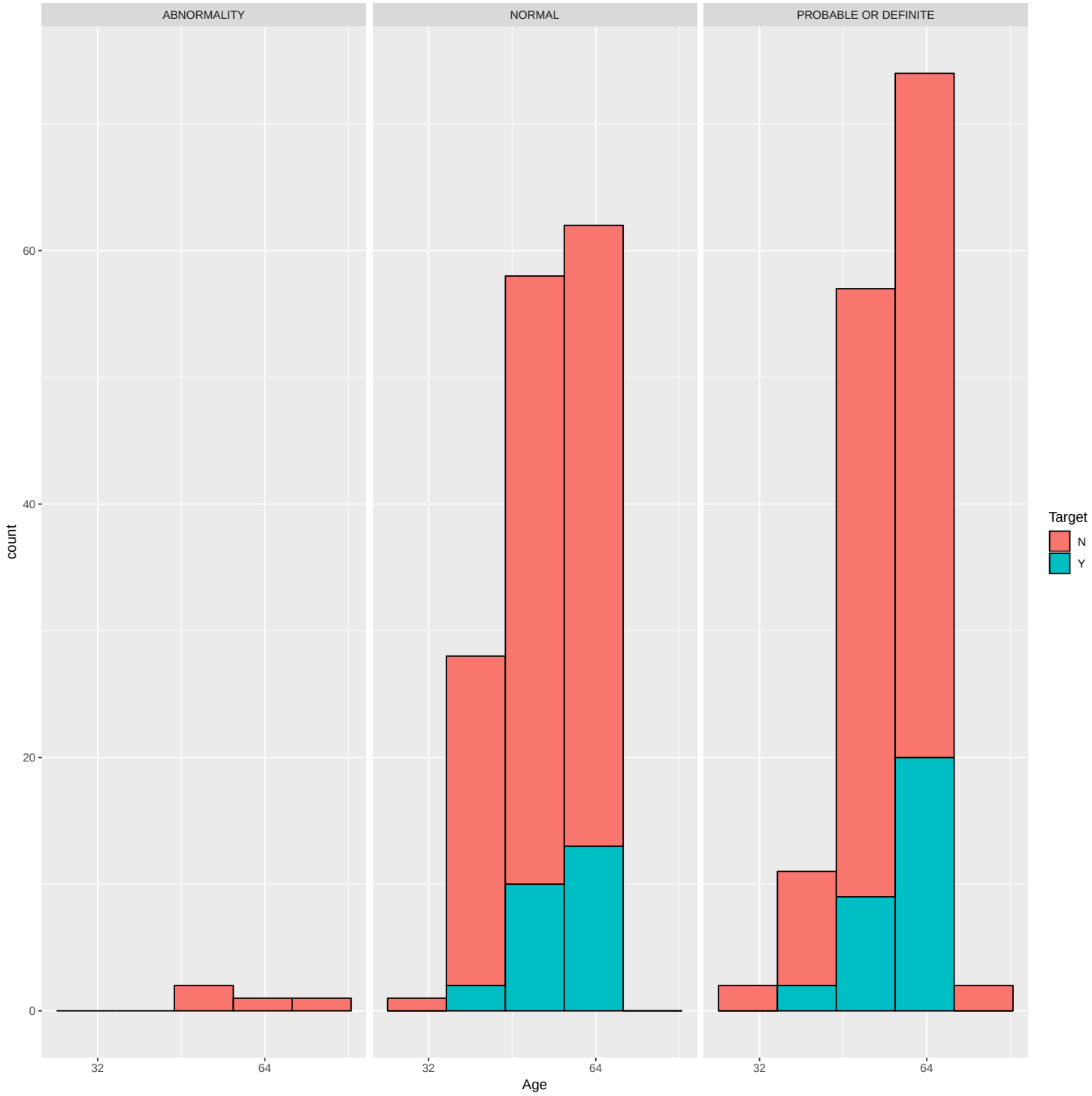
4.1 Age, Chest pain and Target

The stack histogram below shows that chest pain induces heart disease are more prevalent at middle to older age than younger age. Moreover, majority of the patients did not show signs of asymptomatic chest pain while angina pain may not be a good predictor of heart disease.



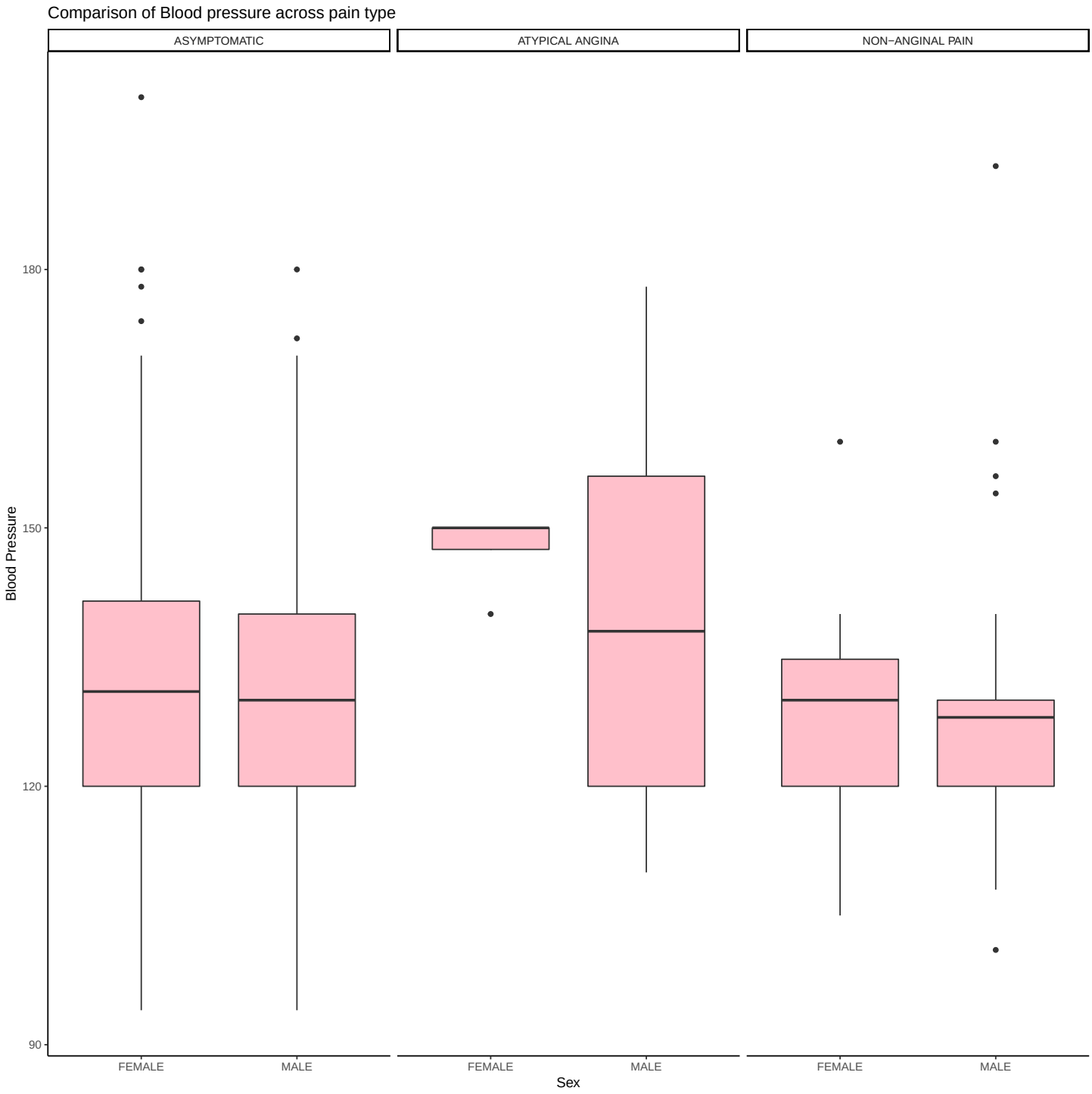
4.2 Age, Rest ECG and Target

The stack histogram below shows that normal and probable RestECG may induces heart disease at older age than younger age.



4.3 Variation of Blood Pressure and Sex across Chest Pain types

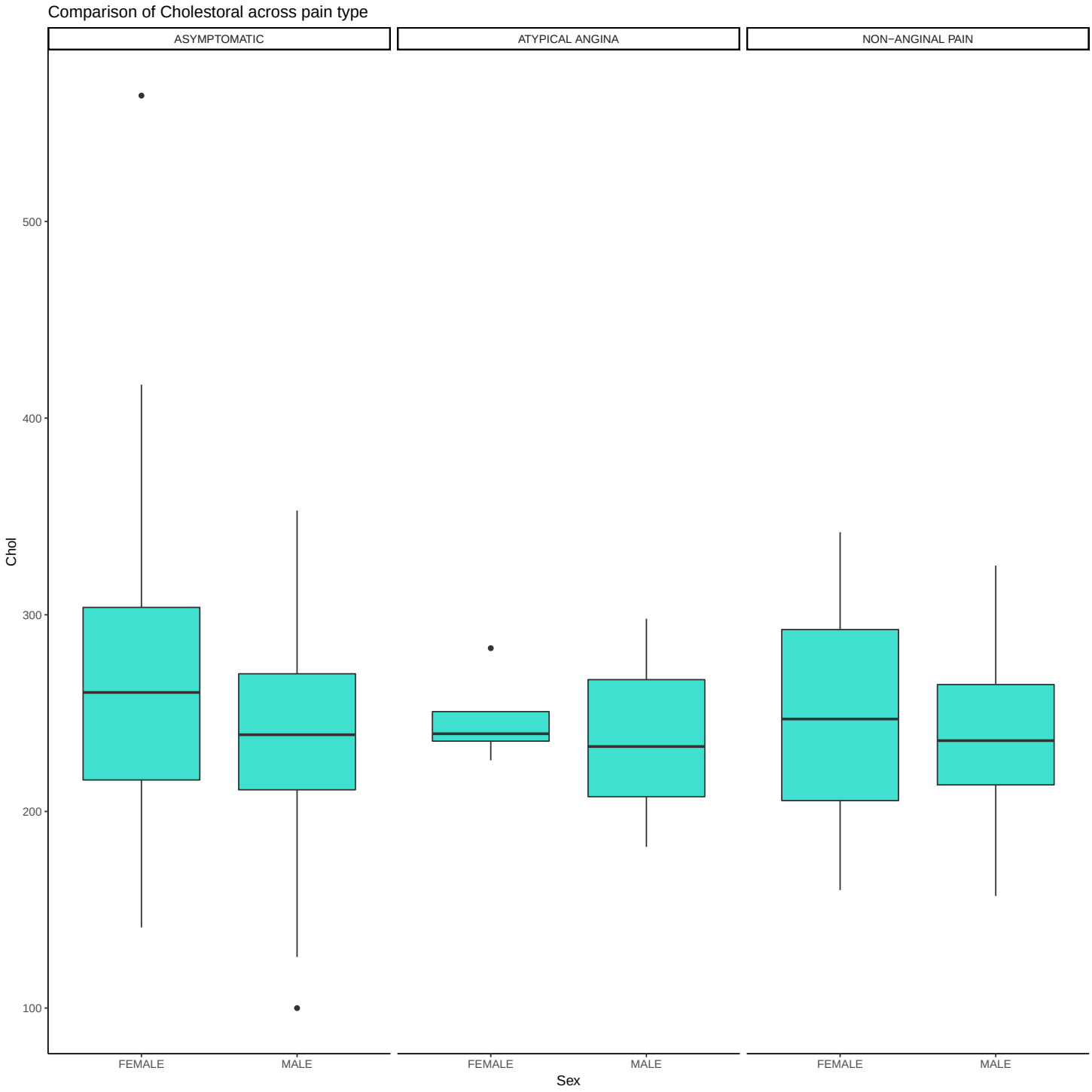
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The box plots above indicate that the mean blood pressure across asymptomatic and non-angina chest pain types are similar irrespective of gender type. However, blood pressure across atypical angina chest pain varies significantly between male and female patients.

4.4 Variation of Cholesterol levels and Sex across chest pain types

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The box plots above indicate that the mean cholesterol levels across asymptomatic and non-angina chest pain types are higher for female than male patients. However, cholesterol levels across atypical angina chest pain are insignificantly higher for female than male patients.

5 Modelling

5.1 Data Processing for Modelling

The highly correlated predictor features (Age, Thalach, Exang and Oldpeak) were dropped from the data set before modeling.

The table below shows the summary of the transformed features before modeling:

Table 2: Feature Summary before Data Modelling

name	type	na	mean	disp	median	mad	min	max	nlevs
Sex	numeric	0	0.765	0.424	1	0.0	0	1	0
CP	numeric	0	3.227	0.939	4	0.0	1	4	0
Trestbps	numeric	0	132.754	18.581	130	14.8	0	200	0
Chol	numeric	0	220.136	93.615	231	54.9	0	603	0
FBS	numeric	0	0.150	0.357	0	0.0	0	1	0
RestECG	numeric	0	0.635	0.840	0	0.0	0	2	0
Target	factor	0	NA	0.482	NA	NA	357	383	2

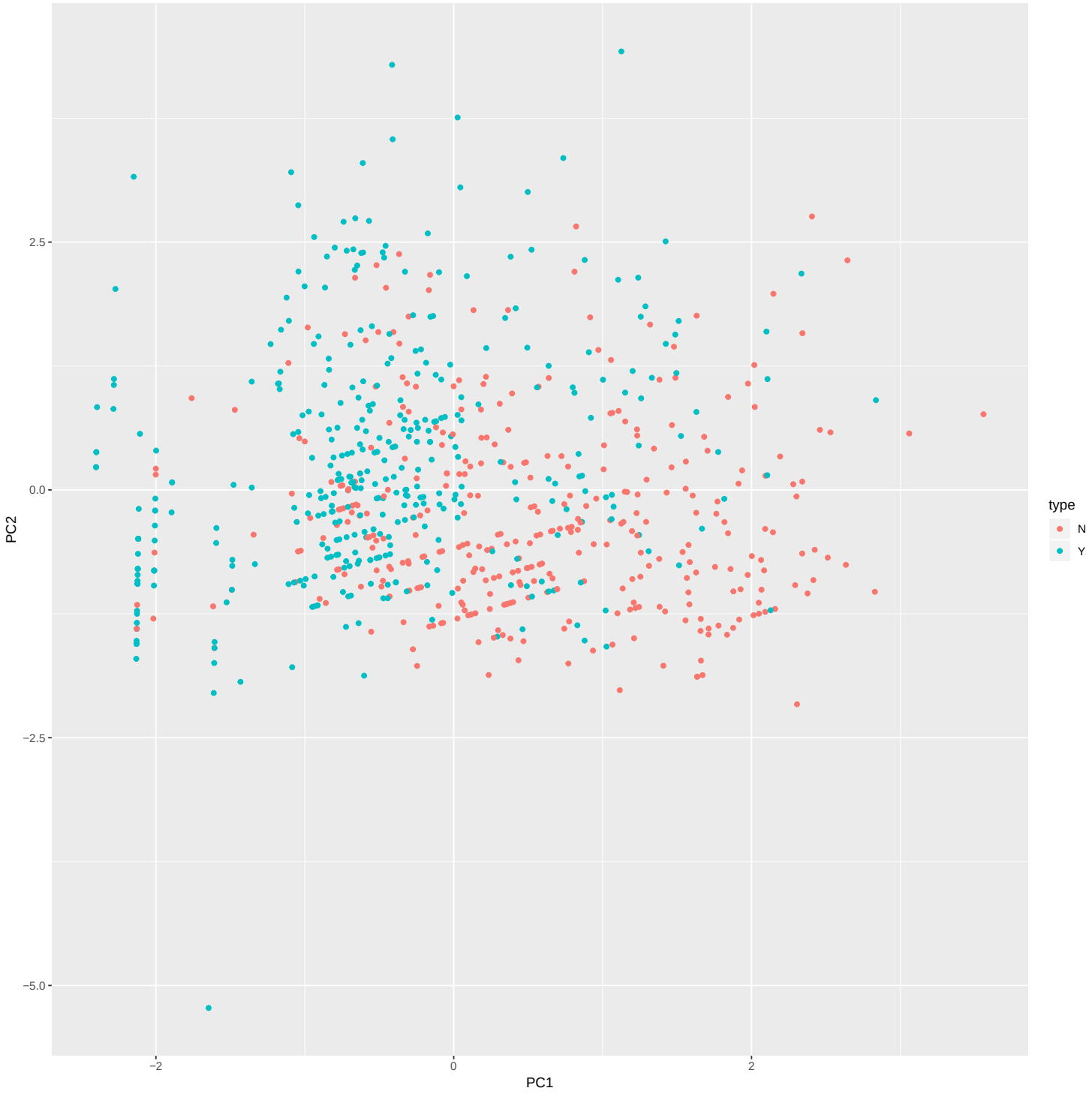
The predictor features were converted into a matrix data type and the target feature was converted into a vector of two levels of factors.

5.2 Principal Component Analysis

The predictor features dataset was scaled and centered. The table below shows the relative importance of the principal components.

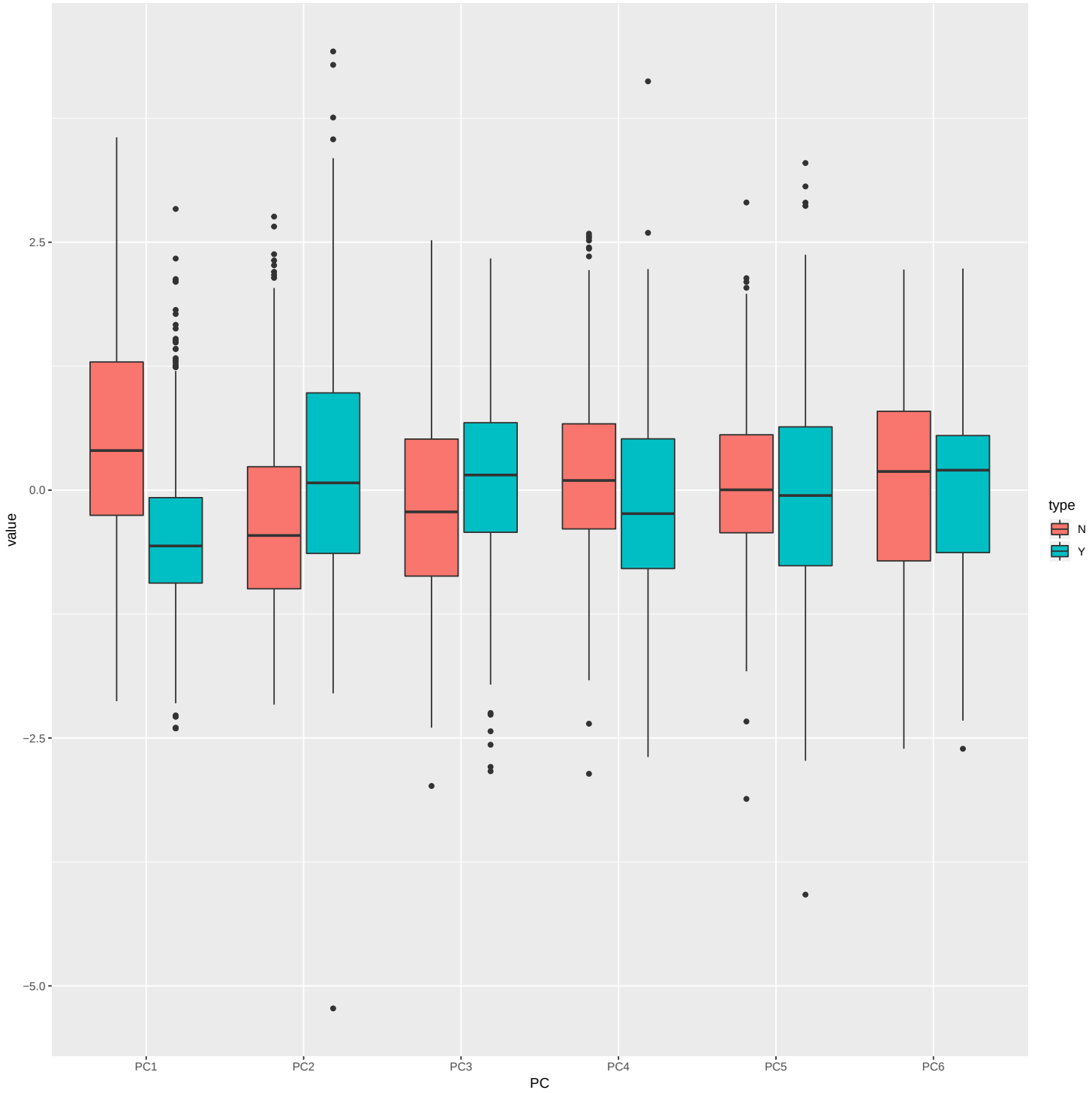
```
## [1] 1
```

```
## Importance of components:
##                           PC1   PC2   PC3   PC4   PC5   PC6
## Standard deviation     1.133 1.129 0.981 0.937 0.908 0.880
## Proportion of Variance 0.214 0.212 0.160 0.146 0.137 0.129
## Cumulative Proportion  0.214 0.427 0.587 0.733 0.871 1.000
```

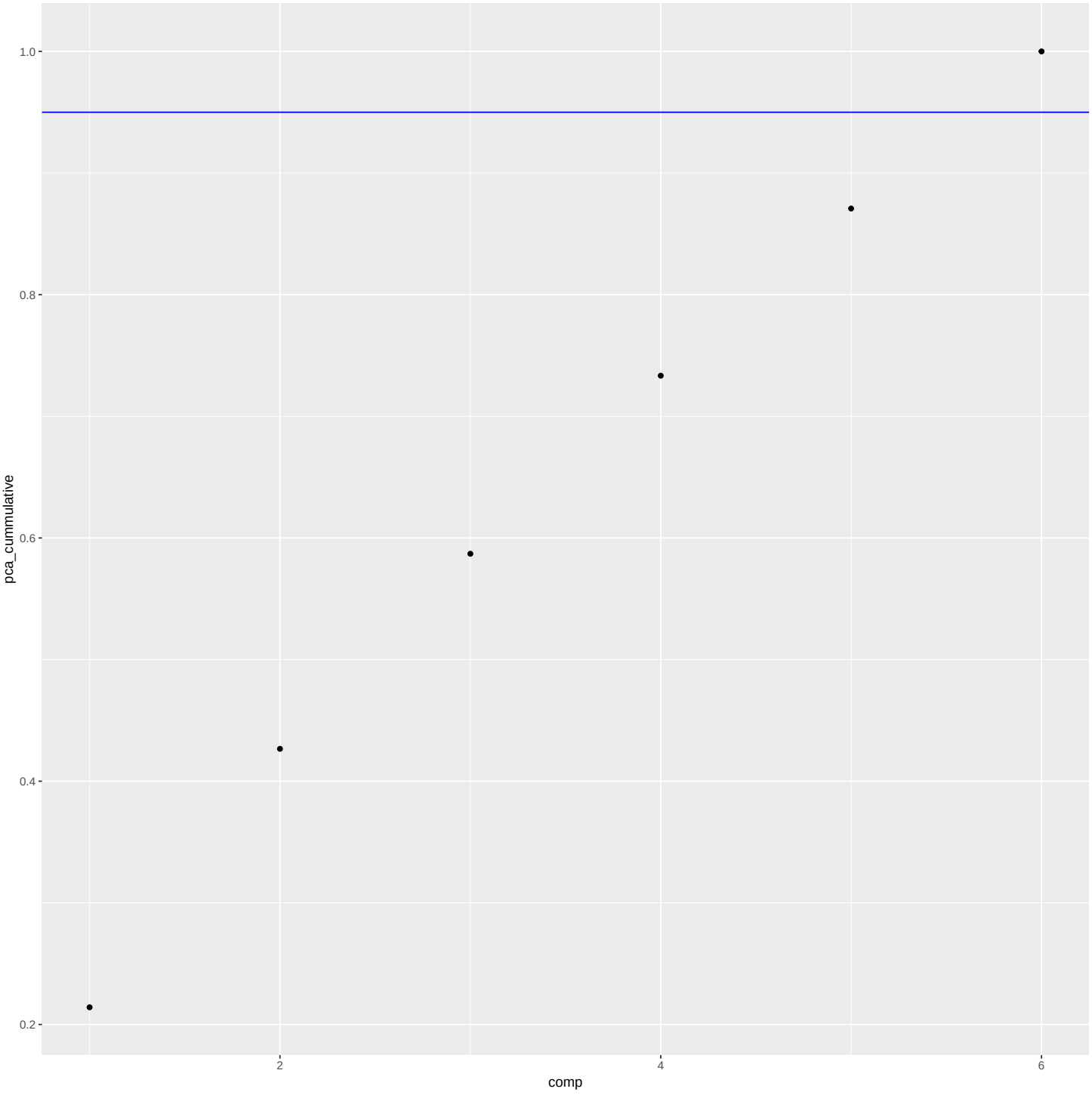


The plot above shows the correlation of first two principal components. Patients with no heart disease tend to have larger values of PC1 than patients with the disease. Also, patients with heart disease tend to have larger values of PC2 than patients with no heart disease.

The box plot below shows all the transformed PCs grouped by the presence or absence of disease. The interquartile ranges overlap for all the PCs. The mean values for all the PCs except PC5 and PC6 are all shifted.



The plot below shows that 95 % of the variance is explained by all principal components in the transformed dataset.



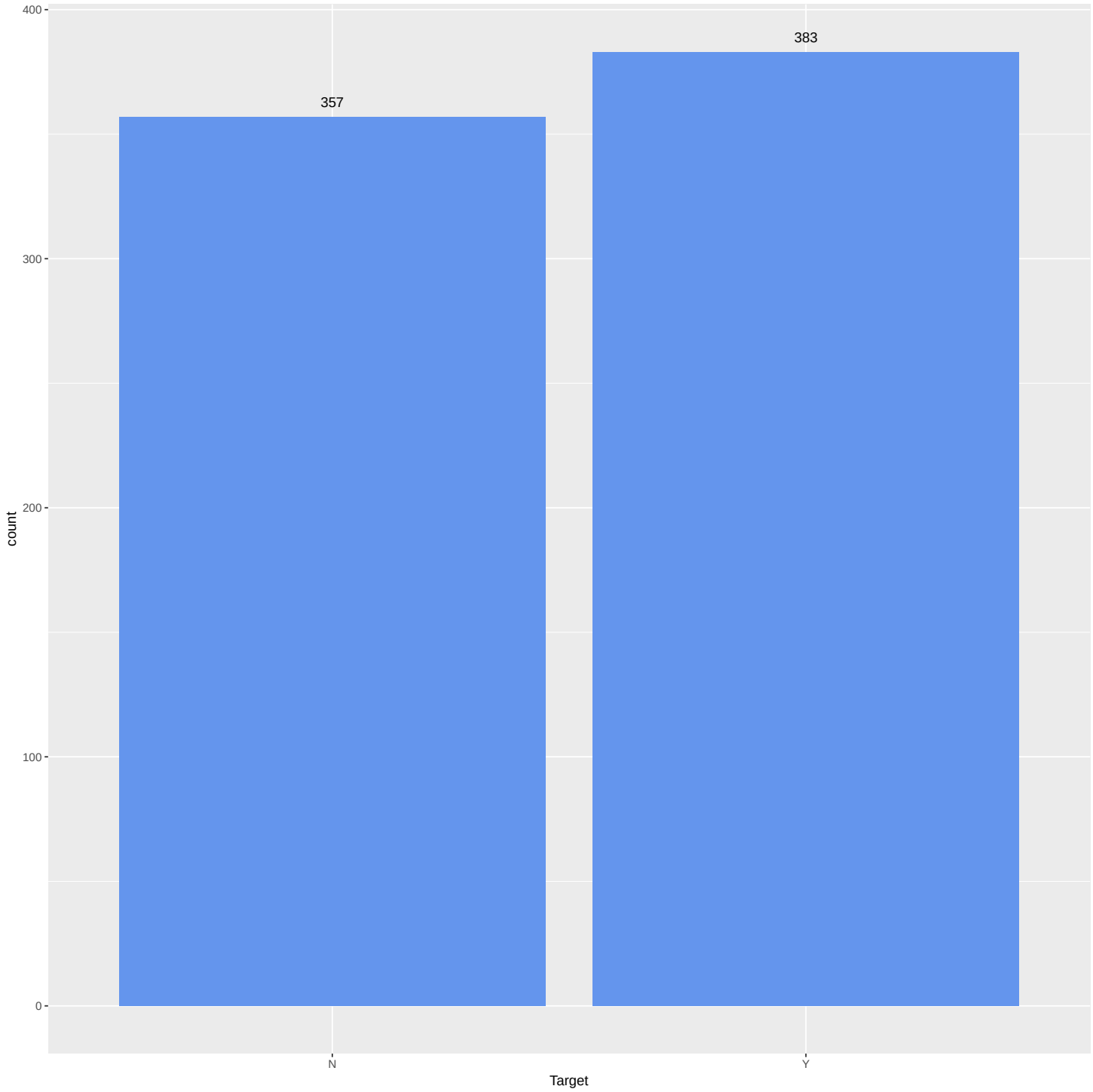
5.3 Data Partitioning

```
#Split into training and test sets
set.seed(1, sample.kind = "Rounding")

test_index <- createDataPartition(hd2$Target, times = 1, p = 0.2, list = FALSE)
test_x <- x_scaled[test_index,]
test_y <- hd2$Target[test_index]
train_x <- x_scaled[-test_index,]
train_y <- hd2$Target[-test_index]
```

The transformed data consisting of one binary target (Y-1 and N-0) variable and six explanatory variables (CP,

Sex, Chol, Trestbps, RestECG and FBS) were split into 80% and 20% as training and test datasets, respectively. Each set resembled the complete data by having the same proportion of target classes as shown in the bar plot below . The modelling was implemented in R with mainly caret and caretEnsemble packages.



6 Methods and Analysis

The following machine learning algorithms were considered in this study:

1. **Kmean clustering**: it is a type of unsupervised algorithm which solves a clustering problem by a simple and easy way to classify a given data through a certain number of clusters. The cluster refers to a collection of data points aggregated together because of certain similarities. The 'means' in the K-means refers to averaging of the data when finding the centroid. The algorithm stops the creating and optimizing of clusters

when the centroids have stabilized or a predefined number of iterations have been achieved. The kmean model was developed using 2 cluster centres to predict weather a patient has heart disease.

```
## Warning in set.seed(3, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
## [1] 0.43
```

The results of the Kmean is represented below:

```
kmean_results <- tibble(method = "K mean model", OverallAccuracy = r1)
kmean_results %>% knitr::kable()
```

method	OverallAccuracy
K mean model	0.43

2. **Logistic regression:** logistic regression is a well-known method in statistics that is used to predict the probability of an outcome, and is especially popular for binary classification tasks. The algorithm predicts the probability of occurrence of an event by fitting data to a logistic function to predict weather a patient has heart disease.

```
## [1] 0.792
```

The results of the logistic regression analysis is represented below:

```
glm_results <- tibble(method = "Logistic Regression model", OverallAccuracy = r2)
glm_results %>% knitr::kable()
```

method	OverallAccuracy
Logistic Regression model	0.792

3. **LDA** : linear discriminant analysis (LDA) is a technique used in machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination is used as a linear classifier. The "lda" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.792
```

The results of the linear descriminant analysis is represented below:

```
lda_results <- tibble(method = "Linear Descriminant Analysis model", OverallAccuracy = r3)
lda_results %>% knitr::kable()
```

method	OverallAccuracy
Linear Descriminant Analysis model	0.792

4. **QDA** : quadratic discriminant analysis (QDA) provides an alternative approach like LDA to logistic regression, the QDA classifier assumes that the observations from each class are drawn from a Gaussian distribution assuming that each class has its own covariance matrix. The "qda" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.772
```

The results of the quadratic descriminant analysis is represented below:

```
qda_results <- tibble(method = "Quadratic Descriminant Analysis model", OverallAccuracy = r4)
qda_results %>% knitr::kable()
```

method	OverallAccuracy
Quadratic Descriminant Analysis model	0.772

5. **Loess gram**: loess short (locally estimated scatterplot smoothing) regression is a non-parametric approach that fits multiple regression in local neighborhood. The fit at a point is made using points in a neighbourhood of, weighted by their distance from the original point. The differences in 'parametric' variables are ignored when computing the distances. The "gamLoess" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## Loading required package: gam
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded gam 1.16.1
```

```
## [1] 0.792
```

The results of the locally estimated scatterplot smoothing is represented below:

```
loess_results <- tibble(method = "Locally Estimated Scatterplot Smoothing model", OverallAccuracy = r
loess_results %>% knitr::kable()
```

method	OverallAccuracy
Locally Estimated Scatterplot Smoothing model	0.792

6. **KNN**: knn or k-nearest neighbors algorithm is one of the simplest machine learning algorithms and is an example of instance-based learning, where new data are classified based on stored, labeled instances. The distance between the stored data and the new instance is calculated by means of some kind of a similarity measure. This similarity measure is typically expressed by a distance measure such as the Euclidean distance, cosine similarity or the Manhattan distance. The number of k nearest neighbors was tuned with a sequence of values ranging between 2 and 21 and the optimal neighbor of 5 was used.

```
## Warning in set.seed(2, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
##    k
## 6 13
```

```
## [1] 0.779
```

The results of the k-nearest neighbors is represented below:

```
knn_results <- tibble(method = "k-nearest neighbors model", OverallAccuracy = r6)
knn_results %>% knitr::kable()
```

method	OverallAccuracy
k-nearest neighbors model	0.779

7. **Random Forest**: an ensemble approach to .finding the decision tree that best fits the training data by creating many decision trees and then determining the "average" one. The "random" part of the term refers to building each of the decision trees from a random selection of features; the "forest" refers to the set of decision trees. The number of variables randomly sampled as candidates at each split (i.e. mtry) was fine-tuned over a sequence of values ranging between 3 and 9 and an optimal value of 3 was used to train the model. The "rf" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.785
```

The results of the random forest is represented below:

```
rf_results <- tibble(method = "Random Forest model ", OverallAccuracy = r7)
rf_results %>% knitr::kable()
```

method	OverallAccuracy
Random Forest model	0.785

8. **Neural network with PCA**: a single-hidden-layer neural network, possibly with skip-layer connections that has the ability to 'learn' relationships among variables. They represent an innovative technique for model fitting that doesn't rely on conventional assumptions necessary for standard models and they can also quite effectively handle multivariate response data. A neural network model is very similar to a non-linear regression model, with the exception that the former can handle an incredibly large amount of model parameters. For this reason, neural network models are said to have the ability to approximate any continuous function. The model was preprocessed by centering and scaling before "pca" was applied. The model was also tuned with a tuneLength of 10 before prediction on the test set.

```
## [1] 0.792
```

The results of the neural network with PCA is represented below:

```
nnet_pca_results <- tibble(method = "Neural Network with PCA model", OverallAccuracy = r8)
nnet_pca_results %>% knitr::kable()
```

method	OverallAccuracy
Neural Network with PCA model	0.792

9. **MARS**: multivariate additive regression splines (MARS) is a none-parametric method with elements of generalized additive models (GAM) or neuronal networks and regression trees. It does not assume a deterministic relationship (linear, logistic, pareto) between the predictors and the response. However, it uses multiple linear regressions to make its predictions. It essentially creates a piecewise linear model which provides an intuitive stepping block into nonlinearity after grasping the concept of linear regression and other intrinsically linear models. The model was tuned with a tuneLength of 10 and the "earth" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.792
```

The results of the multivariate additive regression splines is represented below:

```
earth_results <- tibble(method = "Multivariate Additive Regression Splines model", OverallAccuracy =
earth_results %>% knitr::kable()
```

method	OverallAccuracy
Multivariate Additive Regression Splines model	0.792

10. **Support Vector Machine with Radial Kernel**: support vector machines (SVMs) are supervised learning models that analyze data and recognize patterns, and that can be used for both classification and regression tasks. It does not use probabilistic model like any other classifier but simply generates hyperplanes or simply putting lines ,to separate and classify the data in some feature space into different regions. A radial kernel with a tuningLength of 15 was used to account for the smoothness of the decision boundary and control the variance of the model.

```
## [1] 0.805
```

The results of the support vector machine with radial kernel is represented below:

```
svm_rd_results <- tibble(method = "Support Vector Machine with Radial Kernel model", OverallAccuracy
svm_rd_results %>% knitr::kable()
```

method	OverallAccuracy
Support Vector Machine with Radial Kernel model	0.805

11. **Adaboost**: the algorithm fits a sequence of weak learners on different weighted training data. It starts by predicting original data set and gives equal weight to each observation. If prediction is incorrect using the first learner, then it gives higher weight to observation which have been predicted incorrectly. Being an iterative process, it continues to add learner(s) until a limit is reached in the number of models or accuracy. The model was tuned with a tuneLength of 5 and the "adaboost" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.718
```

The results of the additive boosting is represented below:

```
adaboost_results <- tibble(method = "Additive Boosting model", OverallAccuracy = r11)
adaboost_results %>% knitr::kable()
```

method	OverallAccuracy
Additive Boosting model	0.718

12. **Decision Tree**: the model is represented as a sequence of branching statements. Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. The "rpart" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.805
```

The results of the decision tree is represented below:

```
dt_results <- tibble(method = "Decision Tree model", OverallAccuracy = r12)
dt_results %>% knitr::kable()
```

method	OverallAccuracy
Decision Tree model	0.805

13. **Extreme Gradient Boost with Dart**: xgboost mostly combines a huge number of regression trees with a small learning rate to optimize booting tree algorithms. Trees added early are significant and trees added late are unimportant. The DART(dropout meet multiple additive regression trees) method was adopted to drop trees in order to resolve overfitting. By employing multi-threads and imposing regularization, xgboost is able to utilize more computational power and get more accurate prediction. The algorithm is slow due to the

randomness introduced during training. The model was tuned with a tuneLength of 5 and the "xgboostDART" function was implemented on the training dataset and prediction was done on the test set to find weather a patient has heart disease.

```
## [1] 0.812
```

The results of the extreme gradient boasting is represented below:

```
xgb_dart_results <- tibble(method = "Extreme Gradient Boosting model", OverallAccuracy = r13)
xgb_dart_results %>% knitr::kable()
```

method	OverallAccuracy
Extreme Gradient Boosting model	0.812

14. **Ensemble using caretEnsemble** : ensemble methods are models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction. The thirteen models described above were all included in this ensemble modelling.

```
## Warning in set.seed(6, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
## [1] 0.782
```

The results of the ensemble is represented below:

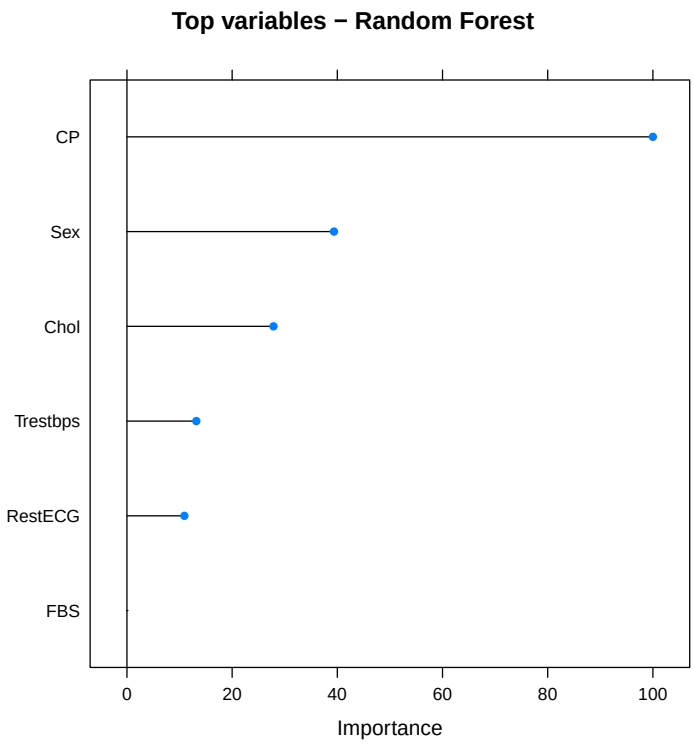
```
ensemble_results <- tibble(method = "Ensemble model", OverallAccuracy = r14)
ensemble_results %>% knitr::kable()
```

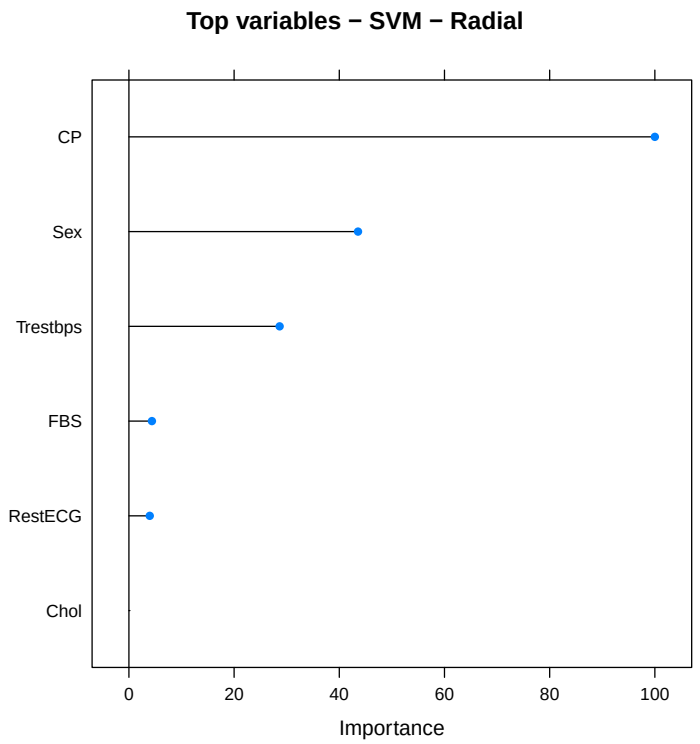
method	OverallAccuracy
Ensemble model	0.782

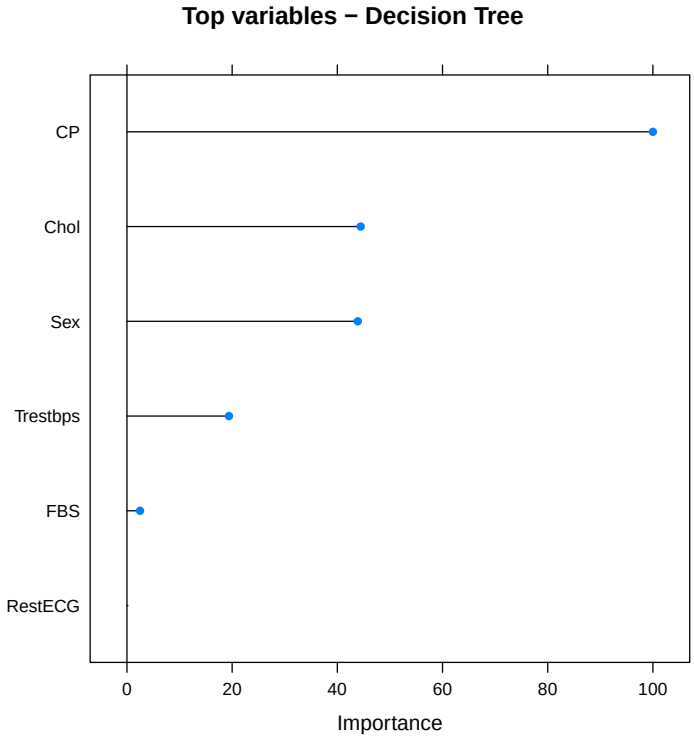
7 Results

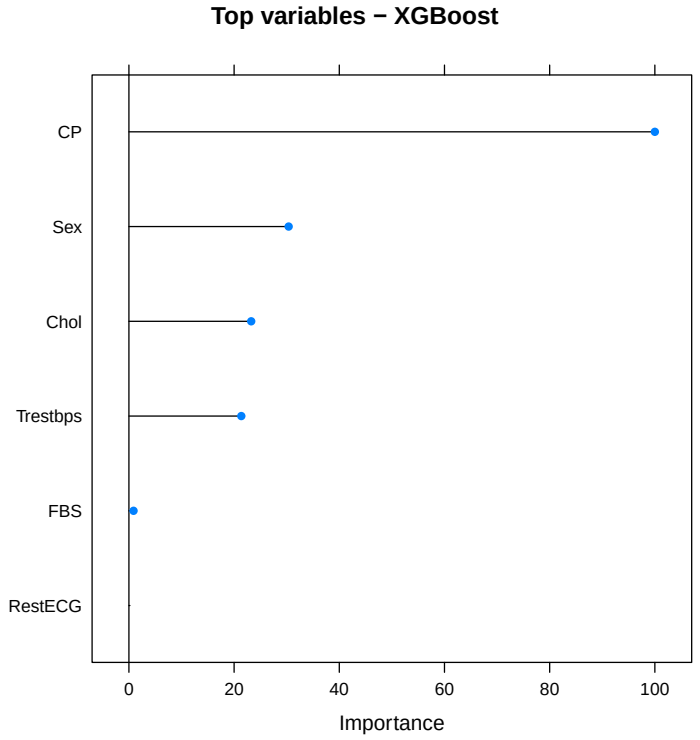
7.1 The Overall Accuracy Results

The plots below show the predictor features of importance for four classifiers, random forest, support vector machines, decision tress and extreme gradient boosting:









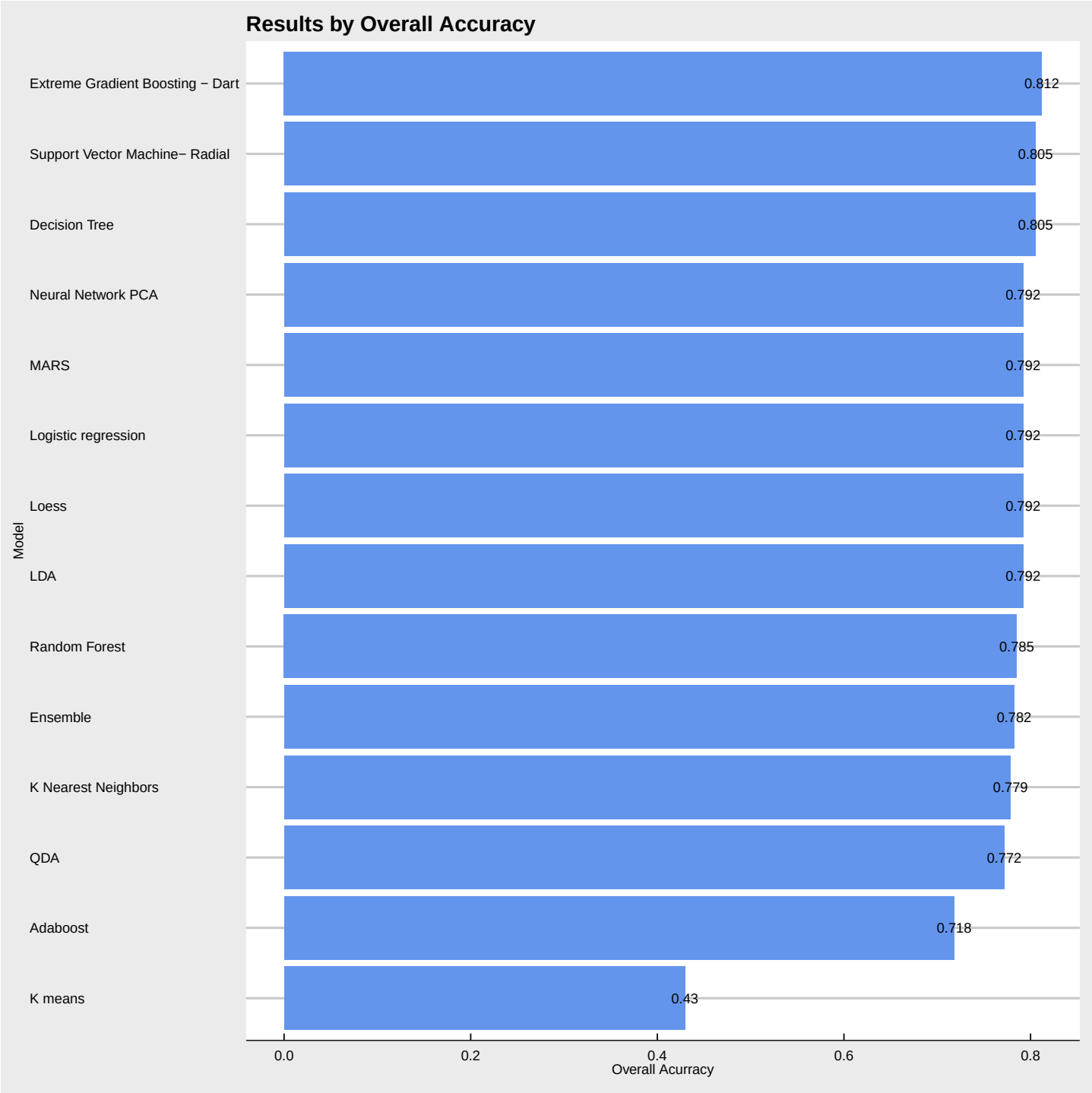
The variables of importance show a consistent trend of chest pain being the most important predictor of heart disease. Other predictors that are important are sex and cholesterol levels although the SVM model ranked "Sex" as the least important predictor. The overall accuracy scores for the used models are shown below:

Table 17: Model Accuracy

Model	Accuracy
K means	0.430
Logistic regression	0.792
LDA	0.792
QDA	0.772

Model	Accuracy
Loess	0.792
K Nearest Neighbors	0.779
Random Forest	0.785
Neural Network PCA	0.792
MARS	0.792
Support Vector Machine- Radial	0.805
Adaboost	0.718
Decision Tree	0.805
Extreme Gradient Boosting - Dart	0.812
Ensemble	0.782

The graphical representation of the models ranked by their overall accuracy is shown below:



The highest overall accuracy score that was computed is 0.812

8 Discussion

The data exploration indicated that patient chest pain, sex, cholesterol levels, resting blood pressure, fasting blood sugar and resting electrocardiogram were potential explanatory variables for predicting the presence of heart disease. The chest pain variable was found to have the highest predictive power. The top three models that performed reasonably well were extreme gradient boosting (xgb), support vector machine with radial kernel and decision tree. The best and the least performed models, xgb and k means reported an overall accuracy score of 81.2% and 43%, respectively.

9 Conclusions

9.1 Summary

Explanatory features, slope of the peak exercise ST segment (Slope), reverse defects (Thal) and the number of major vessels (0-3) colored by fluoroscopy (CA) were dropped out of the dataset because they were missing more than 30% of their measurements. In order to reduce multicollinearity and increase the independency of predictors, explanatory features (age; Age, exercise induced angina ; Exang, maximum heart rate ; Thalach and ST depression induced by exercise relative to rest; Oldpeak) found to have a degree of correlation above 0.75 between variables were also dropped from the dataset.

It can be concluded that, for the UCI heart disease dataset, extreme gradient boosting model was the best to predict whether a patient have heart disease or not.

9.2 Limitations

Overall accuracy has some disadvantages when the dataset is imbalance. However, the target feature in the transformed dataset was pretty balanced with cases of heart disease and no disease being 51.76 % and 48.24 %, respectively. Most of the model parameters were either partially tuned or not tuned as this would require higher computer time. When predicting heart disease, it can be very dangerous for patients if the patients have heart disease and model predicts otherwise. The prediction of False Negatives should therefore be further investigated before selecting the best model for deployment and future investigations.

10 References

1. Shameer K, Johnson KW, et al. Heart. 1-9, 2018.
2. Detrano R, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation, CA, USA
3. Janosi A, Hungarian Institute of Cardiology. Budapest, Hungary
4. Steinbrunn W, University Hospital, Zurich, Switzerland
5. Detrano R, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation, CA, USA
6. http://archive.ics.uci.edu/ml/datasets/heart+disease
7. https://rafalab.github.io/dsbook/